Time-series Scenario Forecasting

Sriharsha Veeramachaneni *

November 14, 2012

Abstract

Many applications require the ability to judge uncertainty of time-series forecasts. Uncertainty is often specified as point-wise error bars around a mean or median forecast. Due to temporal dependencies, such a method obscures some information. We would ideally have a way to query the posterior probability of the *entire* time-series given the predictive variables, or at a minimum, be able to draw samples from this distribution. We use a Bayesian dictionary learning algorithm to statistically generate an ensemble of forecasts. We show that the algorithm performs as well as a physics-based ensemble method for temperature forecasts for Houston. We conclude that the method shows promise for scenario forecasting where physics-based methods are absent.

1 Introduction

Error bars (derived from conditional standard deviation) or predictive intervals (conditional quantiles) attempt to convey the uncertainty in a forecast to the user to enable more-informed decision making. Although such methods work well when forecasting univariate targets (e.g., total rainfall over the next week), they are insufficient for multivariate targets that have complex inter-dependencies (e.g., tomorrow's max temperature, max dew point and total rainfall). The basic problem is that the conditional errors of the forecast are not independent (e.g., if the forecast overestimates tomorrow's max temperature, it is likely to do the same for the dew point.) This is particularly true when forecasting time-series because most variables show strong temporal dependencies.

One approach taken for weather forecasting is to generate several *scenarios* or realizations of the time-series, such that each realization satisfies the dependencies that are known the exist. If the scenario[1] forecasting method simulates draws from the posterior distribution, we may use it to answer complex queries about the forecast (e.g. compute the probability that there will be at least $1''$ of rain and max dew point is $< 80°$,) which is impossible to do with simple error bars or predictive intervals on each of the variables[2].

Two major sources of weather forecast error are the uncertainty in the initial conditions and the incompleteness in the modeling of atmospheric physics. Based on this observation, weather scenario forecasting uses ensemble methods, where different physical models are used to make forecasts, each with several different of perturbations of the initial conditions [1]. The perturbations need to be sensible in their ranges and statistical dependencies. Two commonly used methods to generate perturbations are based on *SVD* (where the perturbations are generated in the direction of singular vectors, thereby accounting for correlations), and *vector breeding* (where the perturbations are iteratively constructed in the most chaotic directions.) The NCEP

*Windlogics Inc., St. Paul, MN (Email: hveera@gmail.com)

[1]We will use "scenario" for and individual forecast and "event" for any property that can be computed from it.

[2]We might not be able to train a system to forecast this particular event directly and use simple error bars because the events that interest the user may be diverse, change frequently or be unknown at training time.

Short-Range Ensemble Forecast (SREF) is one such product where 4 different models are run with several initial conditions [2].

Although ensemble forecasting from multiple models allows the user to see a range of scenarios, it does not provide a way to accurately judge their likelihood. This is because there is no easy way to *a priori* estimate the conditional probability that a particular physical model is the right one. This necessitates a statistical approach. Moreover, when forecasting variables for which there are no good physical models (e.g., wind turbine faults or electric load), a statistical approach may be the only recourse.

Dictionary learning is a statistical method to learn "interesting" directions (or a basis) to compactly summarize high-dimensional data. Dictionary learning with sparse over-complete representations has been applied extensively in image and video processing [3].

We learn a dictionary jointly for the target time-series to be predicted and any available predictor variables, using a recently proposed Bayesian dictionary learning algorithm [4]. At predict time, we draw samples from the conditional distribution of the target time-series given the observed predictors and the dictionary. Each of these samples is a scenario, and from this ensemble of scenarios the probability of any event of interest can be estimated.

2 Statistical Scenario Generation

Let us denote the q dimensional target vector by $\mathbf{y_i}$ (e.g., the time-series of electric load over 84 hours starting at time i), and the corresponding r dimensional predictor vector by $\mathbf{x_i}$ (e.g., the temperature, wind speed and dew point over the same 84 hours). We would like to construct a system to draw samples from the density $P(\mathbf{y}|\mathbf{x})$. Denote the concatenated vector $[\mathbf{x_i}, \mathbf{y_i}]$ by $\mathbf{z_i}$

We model $\mathbf{z_i}$ by the hierarchical model described in [4], which we repeat here for the sake of completeness.

$$\begin{aligned}
\mathbf{z_i} &= \mathbf{D}\mathbf{w_i} + \mathbf{e_i} \\
\mathbf{w_i} &= \mathbf{b_i} \odot \mathbf{s_i} \\
\mathbf{d_k} &\sim \mathcal{N}(0, 1/(q+r)\mathbf{I}) \\
\mathbf{s_i} &\sim \mathcal{N}(0, \gamma_s^{-1}\mathbf{I}) \\
\mathbf{b_{i,k}} &\sim \text{Bernoulli}(\pi_k) \\
\pi_k &\sim \text{Beta}(a/K, (K-1)b/K) \\
\mathbf{e_i} &\sim \mathcal{N}(0, \gamma_e^{-1}\mathbf{I}) \\
\gamma_s &\sim \text{Gamma}(c, d) \\
\gamma_e &\sim \text{Gamma}(e, f)
\end{aligned} \tag{1}$$

where K is the number of dictionary atoms, $\mathbf{d_k}$ is the k^{th} atom, t is the dimension of $\mathbf{z}$, $\odot$ is the element-wise product, and a, b, c, d, e, f are hyper-parameters. Note that the $(q+r) \times K$ dictionary matrix $\mathbf{D}$ can be partitioned into the $q \times K$ dictionary denoted $\mathbf{D_y}$ for the targets and the $r \times K$ dictionary denoted $\mathbf{D_x}$ for the predictors.

The parameters needed to generate data according to the model are given by $\Theta = (\mathbf{D}, \pi, \gamma_\mathbf{s}, \gamma_\mathbf{e})$. Given a training data, the model parameters can be estimated via Gibbs sampling (the formulae for which are derived in [4]) yielding the estimate $\hat{\Theta}$.

At predict time, we approximate $P(\mathbf{y}|\mathbf{x})$ as shown below. (In the equations below note that we are implicitly conditioning on the training data.)

$$\begin{aligned}
P(\mathbf{y}|\mathbf{x}) &= \int_{\Theta,\mathbf{s},\mathbf{b}} P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}, \mathbf{s}, \mathbf{b}) \mathbf{P}(\mathbf{\Theta}, \mathbf{s}, \mathbf{b}|\mathbf{x}) \\
&= \int_{\Theta,\mathbf{s},\mathbf{b}} P(\mathbf{y}|\mathbf{\Theta}, \mathbf{s}, \mathbf{b}) \mathbf{P}(\mathbf{\Theta}, \mathbf{s}, \mathbf{b}|\mathbf{x}) \\
&= \int_{\Theta,\mathbf{s},\mathbf{b}} P(\mathbf{y}|\mathbf{\Theta}, \mathbf{s}, \mathbf{b}) \mathbf{P}(\mathbf{s}, \mathbf{b}|\mathbf{x}, \mathbf{\Theta}) \mathbf{P}(\mathbf{\Theta}|\mathbf{x}) \\
&\approx \int_{\mathbf{s},\mathbf{b}} P(\mathbf{y}|\hat{\mathbf{\Theta}}, \mathbf{s}, \mathbf{b}) \mathbf{P}(\mathbf{s}, \mathbf{b}|\mathbf{x}, \hat{\mathbf{\Theta}})
\end{aligned} \tag{2}$$

where the second line is because the individual components of $\mathbf{z}$ are generated i.i.d., when Θ, $\mathbf{s}$ and $\mathbf{b}$ are known, and for the fourth line we make the approximation $P(\Theta|\mathbf{x}) \approx \delta(\mathbf{\Theta} - \hat{\mathbf{\Theta}})$ (this is reasonable because the one test example will not alter the posterior distributions from those conditioned solely on the training data.)

Therefore in order to draw samples from the posterior distribution, we generate several samples of $\mathbf{s}$ and $\mathbf{b}$ from their posterior distribution given the estimated parameters and the predictor variables, and use each realization to reconstruct the target part of $\mathbf{z}$. The vectors $\mathbf{s}$ and $\mathbf{b}$ are drawn iteratively by Gibbs sampling just as was done during training. From given a sample $\mathbf{s}$ and $\mathbf{b}$ we generate a scenario for the targets as $\hat{\mathbf{y}} = \mathbf{D}_{\mathbf{y}}(\mathbf{s} \otimes \mathbf{b})$. Note that the noise part of the target is not included in the realization.

3 Experiments & Discussion

We evaluate our statistical scenario generation method by comparing to the SREF scenarios for the temperature in Houston. (We could have experimented with other variables such as electric load, but lacking physics-based scenarios as a baseline, the metrics would not have made intuitive sense.) Each instance of the target is the temperature over 84 hours starting at 03Z of a particular day, and each scenario is a 03Z temperature forecast extending out to 84 hours. We had training data for approximately 600 days of temperature, over which we performed 10 fold cross-validation for the statistical approach. We chose $K = 100$ and a burn-in $= 100$ for training and prediction.

The SREF ensemble comprises the 21 combinations of physical models and initial conditions: *em.ctl*, *em.n1*, *em.n2*, *em.p1*, *em.p2*, *eta.ctl1*, *eta.ctl2*, *eta.n1*, *eta.n2*, *eta.p1*, *eta.p2*, *nmm.ctl*, *nmm.n1*, *nmm.n2*, *nmm.p1*, *nmm.p2*, *rsm.ctl1*, *rsm.n1*, *rsm.n2*, *rsm.p1*, *rsm.p2*. We chose the 4 models *em.ctl*, *eta.ctl1*, *nmm.ctl* and *rsm.ctl1* as the predictors. This simulates a situation such as forecasting electric load from a few different weather models.

To be exactly comparable to the SREF scenarios, we generate 21 scenarios for each run-time from the statistical method. Various metrics are used to measure the efficacy of the statistical method. If the scenario were being used to compute the probability of certain events we might evaluate the scenarios for that express purpose. We, however, wanted to evaluate the scenario forecasting method agnostic to the events that might be of interest.

A good metric to evaluate a scenario forecast would estimate the likelihood of observing the observed target values given the distribution of the scenarios. Because this is difficult to do from a finite sample of scenarios, two different metrics that measure the *calibration* and *sharpness* of the scenarios. Calibration measures whether the target is drawn from the same distribution as the scenarios, and sharpness measures how tightly the scenarios cluster around the target (see [6] for a detailed discussion of the desiderata for probabilistic forecasts).

One metric used to judge the calibration for univariate scenarios is the rank histogram, which measures the flatness of the histogram of the ranks of the actual targets among the scenarios. This is difficult to generalize to multivariate targets (as is our lot). For low dimensional problems, one method is the *minimum spanning tree distance rank histogram* [5]. Here the length of the Euclidean minimum spanning tree is computed for the scenarios, as well as for each graph where one of the scenarios is replaced with the actual target. The flatness of the rank of the first length among all the lengths is a measure of how well-calibrated the scenarios are. Figure 1 shows the MSP rank histograms for the SREF and the statistical method. It appears that neither is very well calibrated even if our statistical method is an improvement. (It is possible that the MSP rank histogram method is not suitable for such high dimensional problems as ours.)

We might expect from a scenario forecasting system that at least one of the possible scenarios that are forecast ends up being close to the truth. Therefore we may compare two scenario generation methods by comparing the *closest* scenarios to the actual target. (Clearly this makes sense only when the two methods produce the same number of scenarios.)

Figures 2, 3 and 4 show the RMSE, MAE and Bias of the closest scenario respectively, as function of the forecast horizon. It appears that the SREF models tend to over-forecast the temperature at night and under forecast at midday.

Figure 5 shows the scatter plot of the normalized Euclidean distance of the actual target to

the convex hull of the scenarios for the statistical method against that of SREF. The statistical method is slightly worse owing to a few instances where the actual temperature time-series were far from all the scenarios (see Figure 11).

We measure the sharpness of the scenario forecasts by the Euclidean distance between the two furthest members of the ensemble. Figure 6 shows the scatter plot of the sharpness of the statistical method against the SREF for each of the approximately 600 instances. We note that the statistical ensemble is in general less sharp than the SREF. There is a trade-off between sharpness and calibration, and we expect that with a larger training set we can improve sharpness for a given calibration.

Figures 7 through 15 show the scenarios generated by SREF and the statistical method for various run-times.

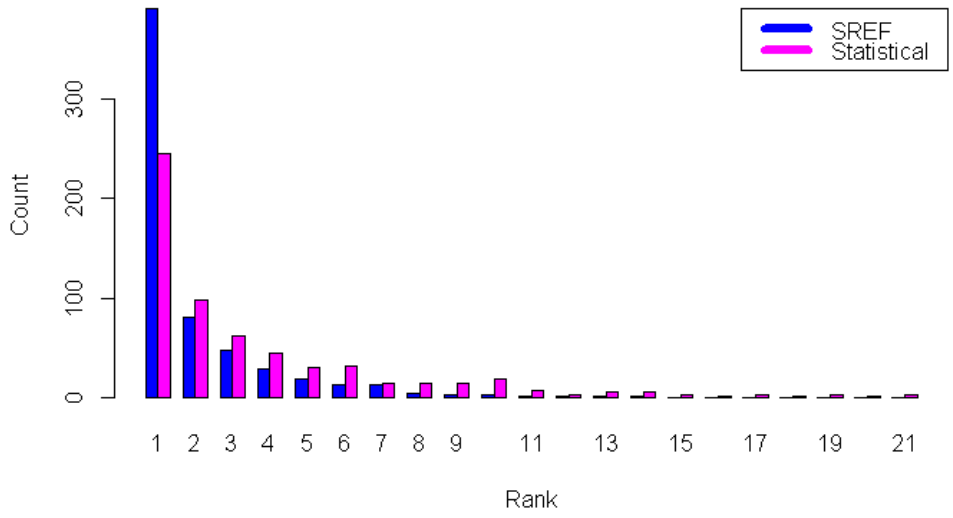

Figure 1: Minimum spanning tree rank histogram.

4 Conclusion and Future Work

The preliminary experimental results presented above show that the proposed method holds promise for time-series scenario forecasting. Although our statistical approach performs about the same or perhaps slightly worse than the physics-based method, it has the advantages that it can be better calibrated, enables a computationally cheap way to generate a large number of scenarios, and can be applied to variables for which physical models are absent. The main disadvantages are that, being a statistical approach, it is limited by the scenarios observed in the training data, and the possibility that the scenarios violate known physical constraints.

Avenues for future work include

- Scenario forecasting a combination of two or more weather variables.
- Experimenting with scenarios over patches of time-series (i.e., sliding windows) to exploit self similarity. Such methods have shown promise in image de-noising and interpolation.
- Modifying the hierarchical model to incorporate prior knowledge about temporal and spatial correlations.
- Studying the impact of training data size on learning high dimensional dictionaries for scenario forecasting.

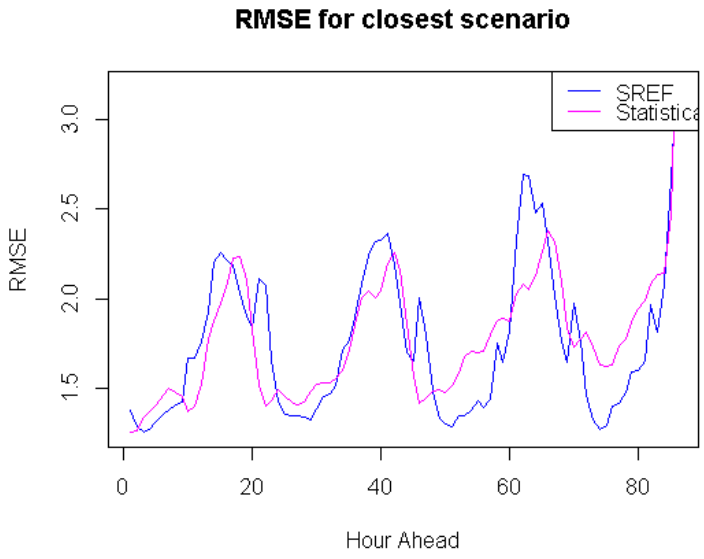

Figure 2: RMSE of the closest scenario by forecast horizon.

References

[1] `http://en.wikipedia.org/wiki/Ensemble_forecasting`

[2] Du, J., et. al., *NCEP short-range ensemble forecast (SREF) system upgrade in 2009*, 19th Conf. on Numerical Weather Prediction and 23rd Conf. on Weather Analysis and Forecasting, 2009.

[3] Bach, F., Mairal, J., Ponce J., Sapiro, G., *Tutorial on Sparse Coding and Dictionary Learning for Image Analysis*, International Conference on Computer Vision and Pattern Recognition, 2010.

[4] Zhou, M. et. al., *Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images*, IEEE Trans. Image Processing, Vol. 21, pp. 130-144, Jan. 2012.

[5] Smith, L. A., Hansen, J. A., *Extending the Limits of Ensemble Forecast Verification with the Minimum Spanning Tree*, Monthly weather review, 132 (6). pp. 1522-1528, 2004.

[6] Gneiting, T. et. al. *Probabilistic forecasts, calibration and sharpness*, Journal of the Royal Statistical Society Series B: Statistical Methodology, 69, 243-268, 2007.

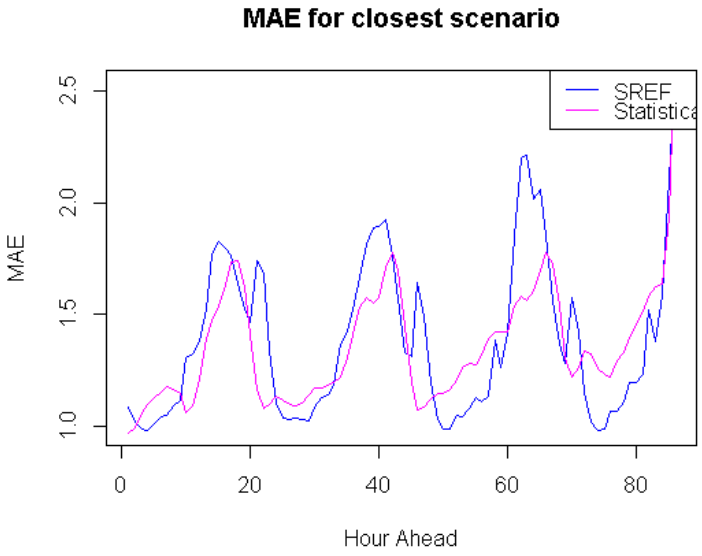

Figure 3: MAE of the closest scenario by forecast horizon.

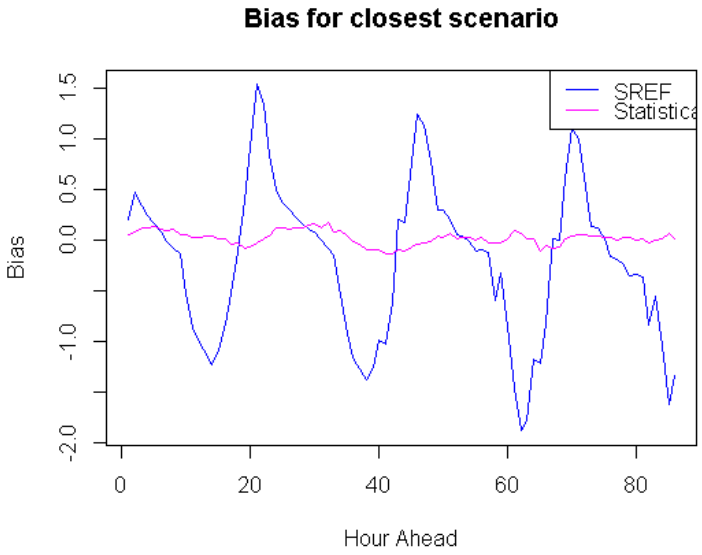

Figure 4: Bias of the closest scenario by forecast horizon.

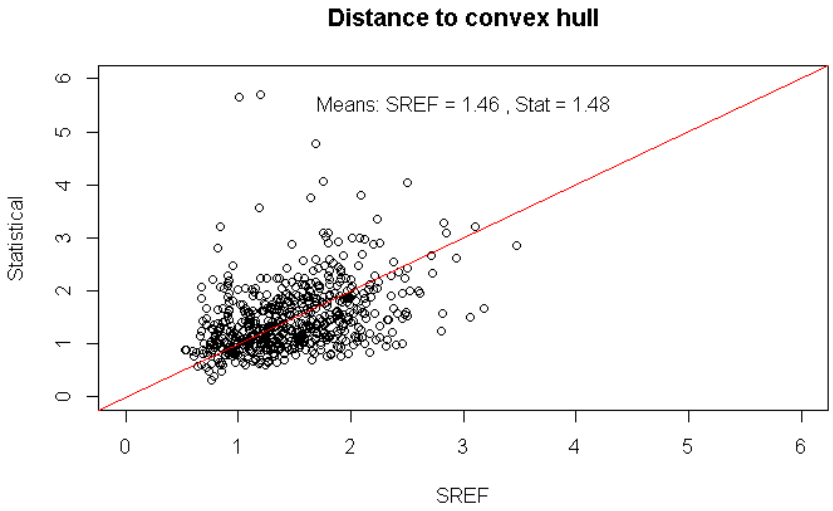

Figure 5: Distance of the actual time-series to the convex hull of the scenarios.

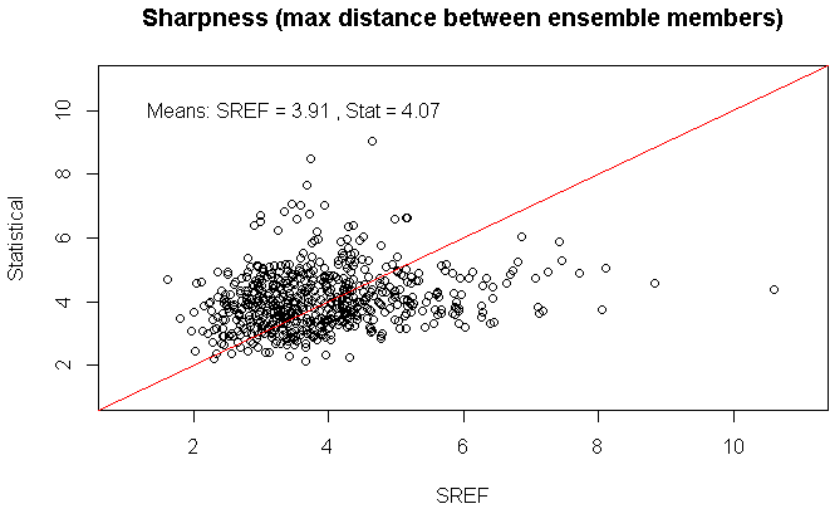

Figure 6: Sharpness SREF vs statistical.

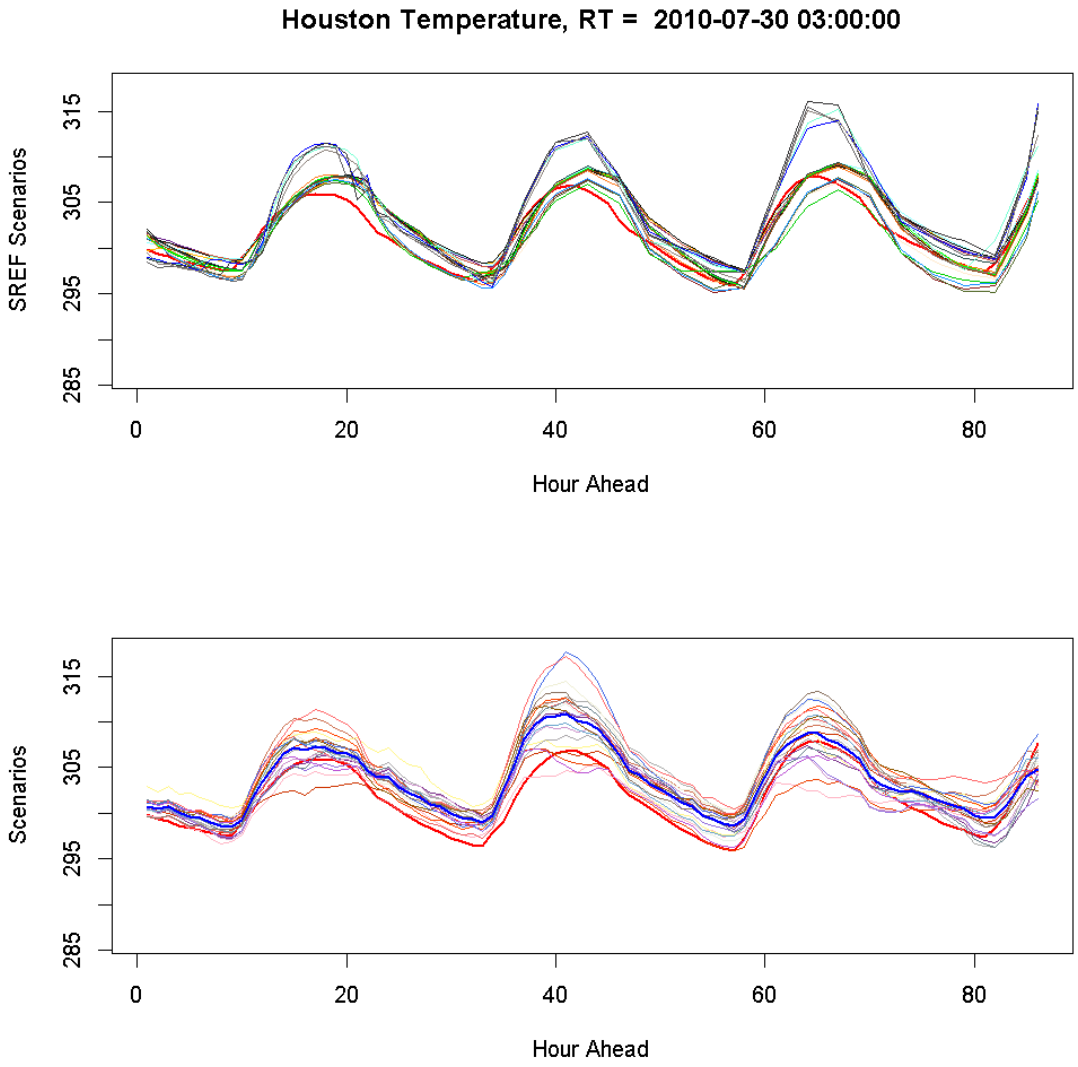

Figure 7: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

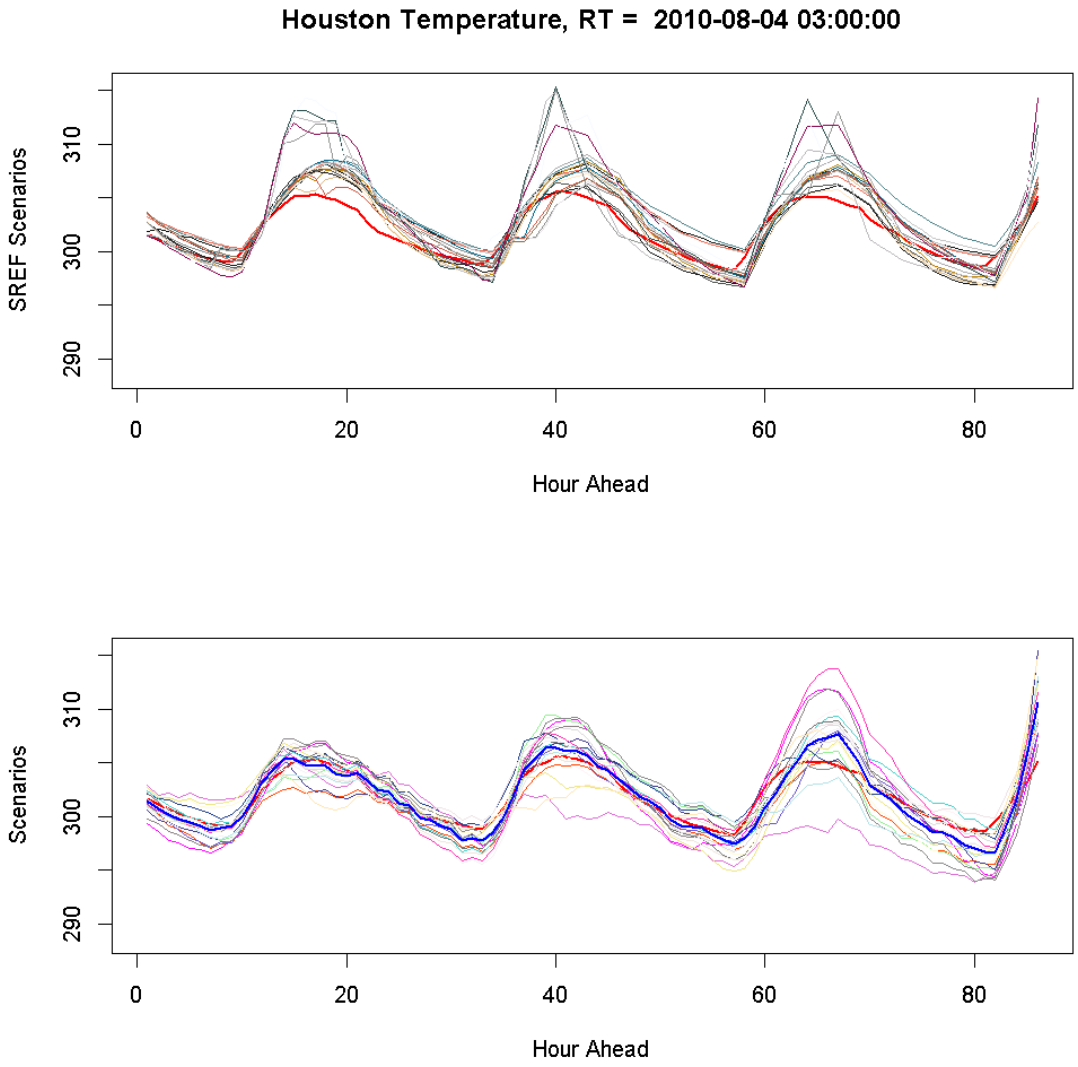

Figure 8: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

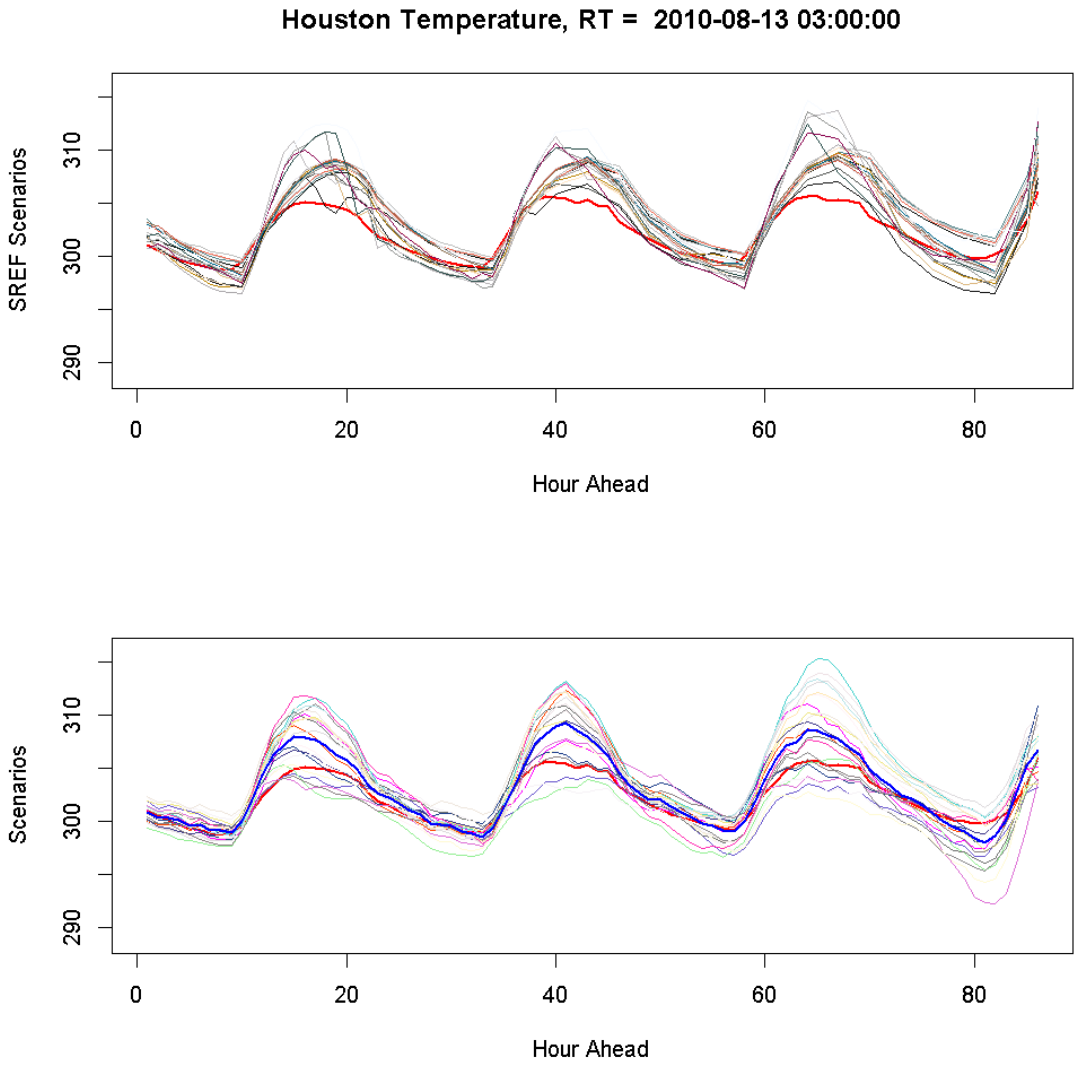

Figure 9: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

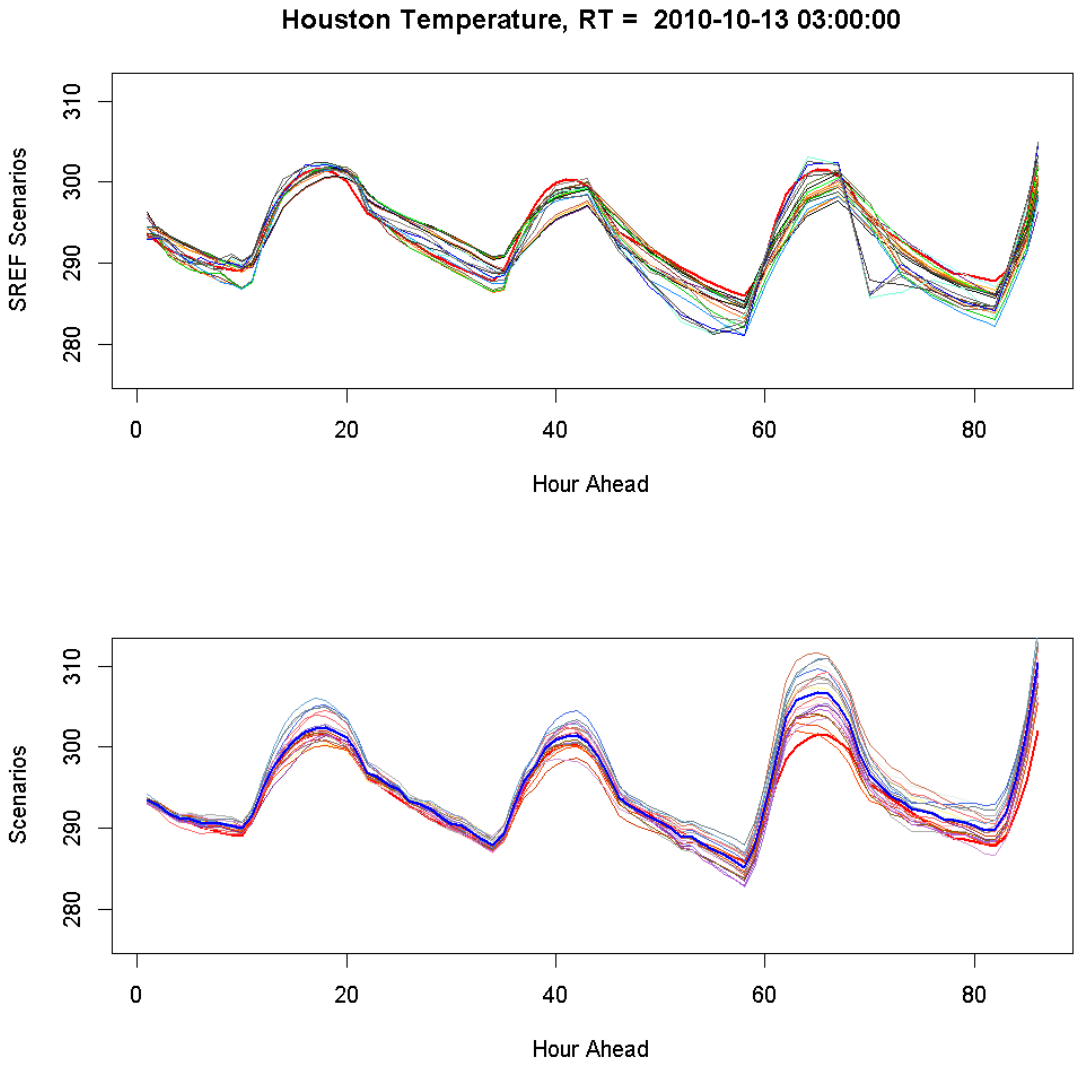

Figure 10: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

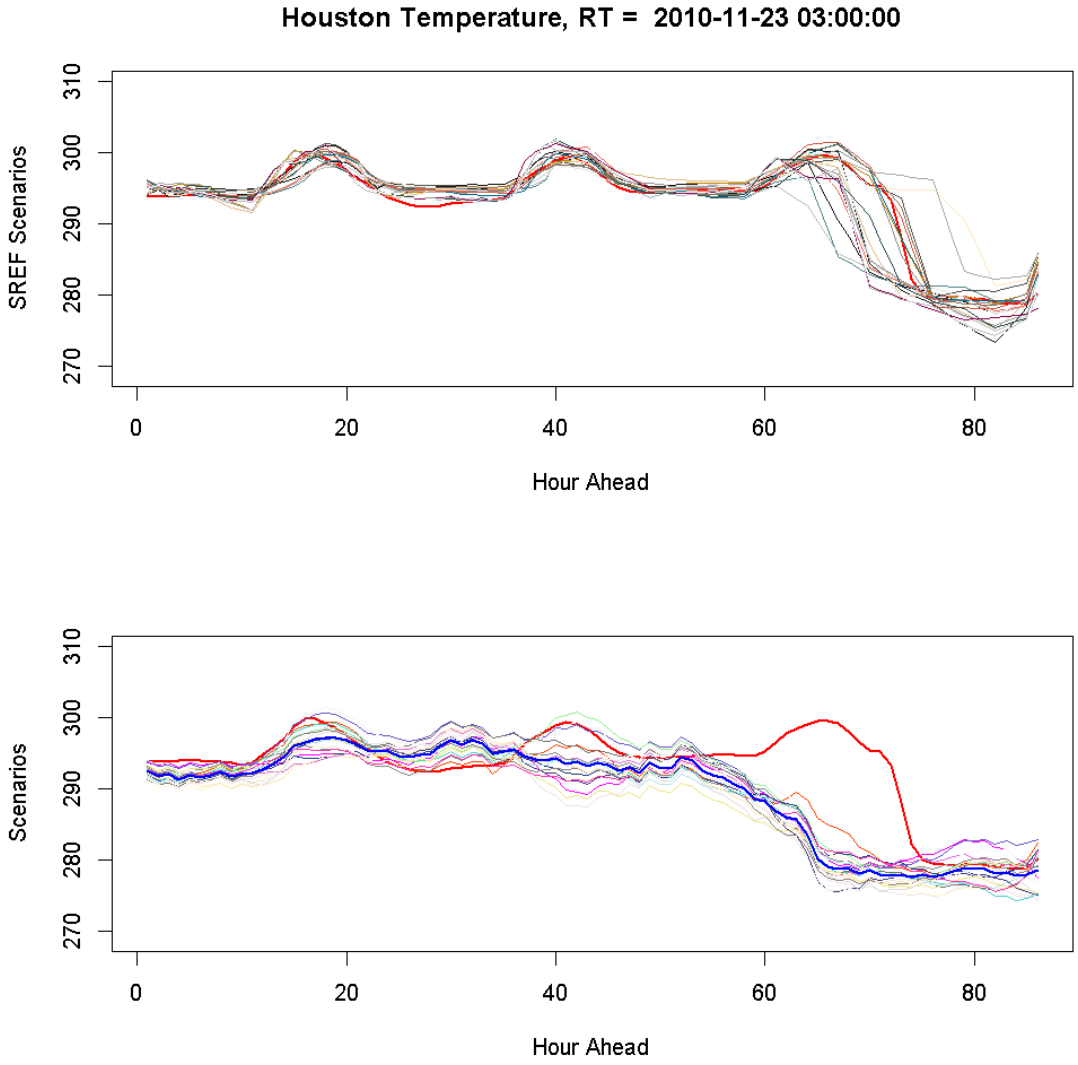

Figure 11: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

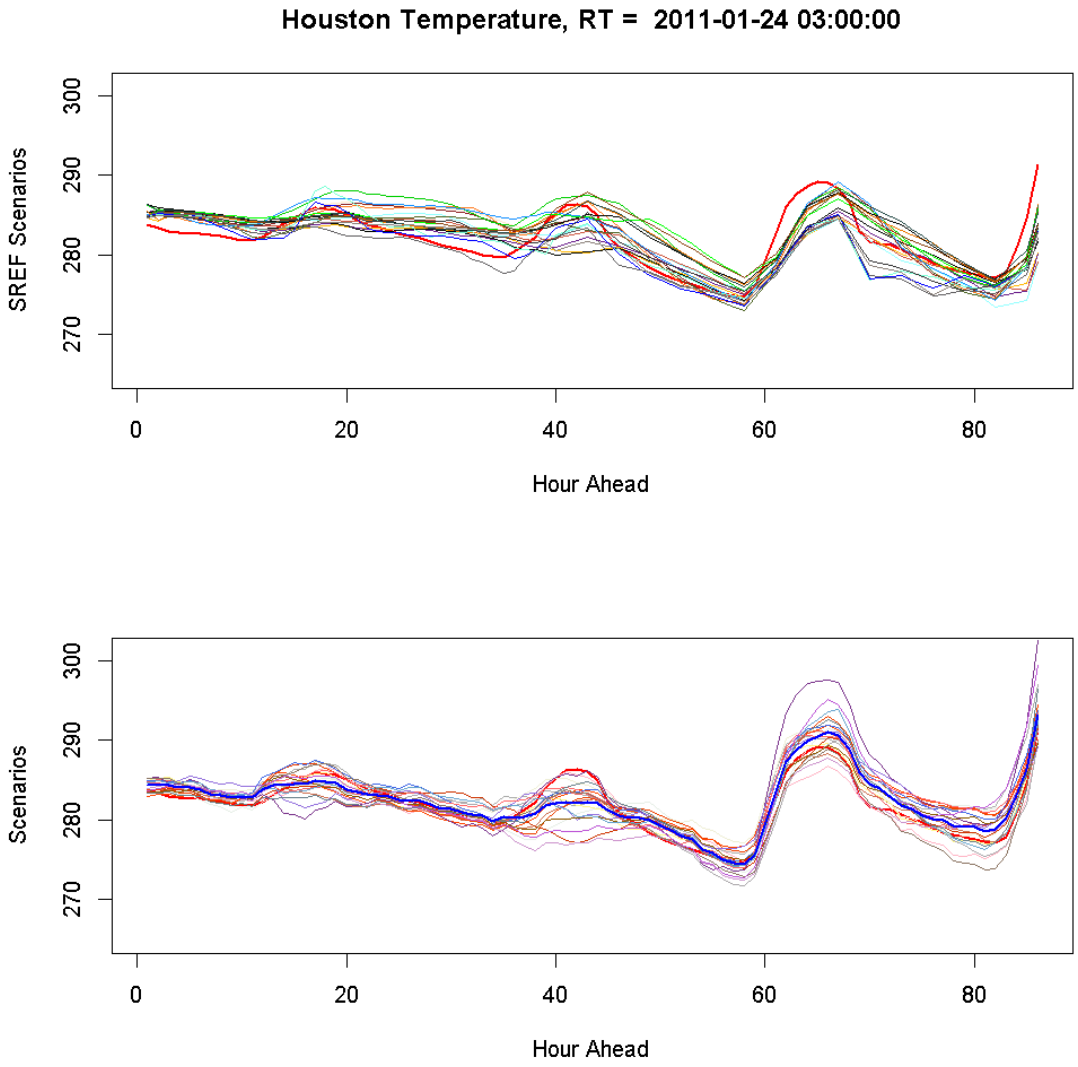

Figure 12: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

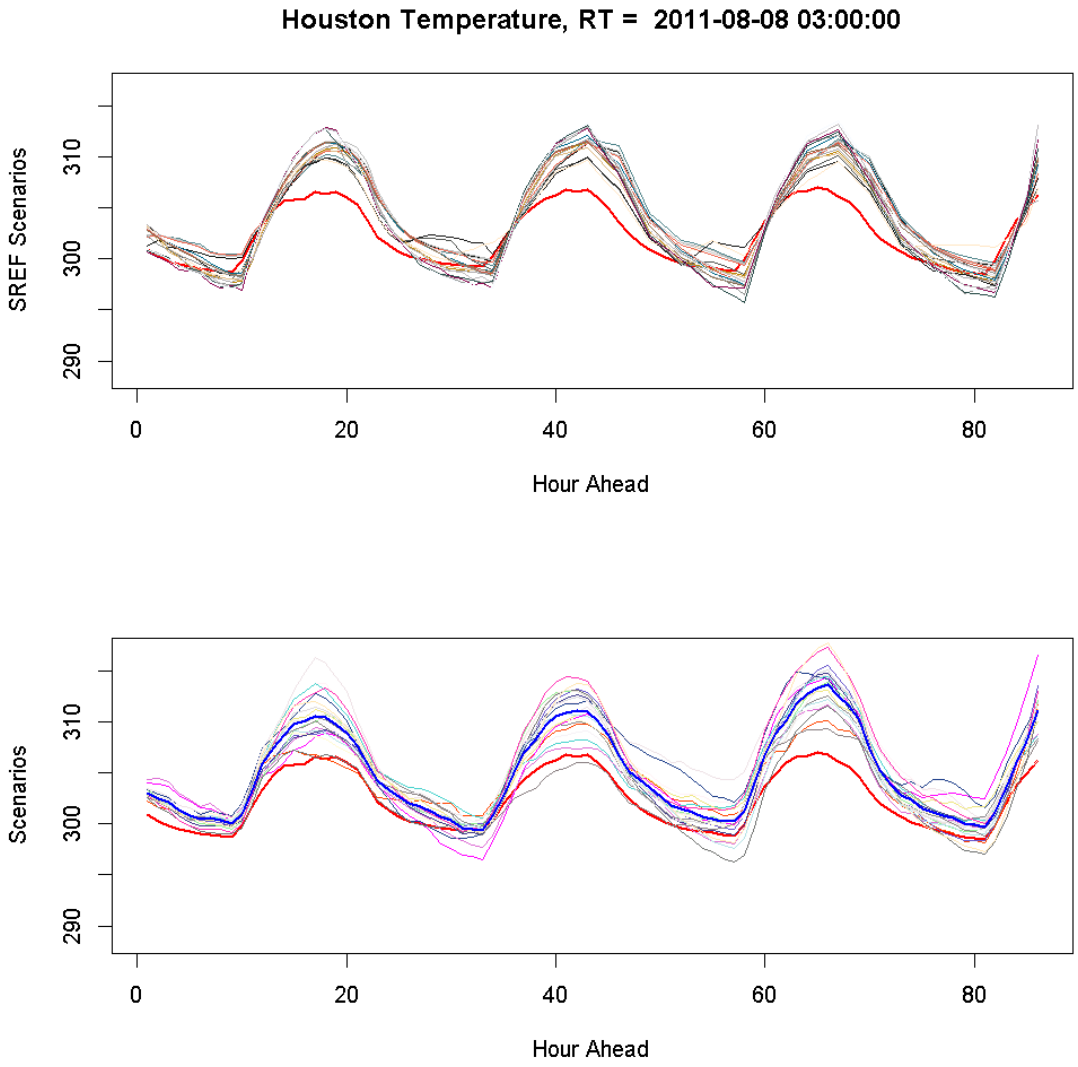

Figure 13: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

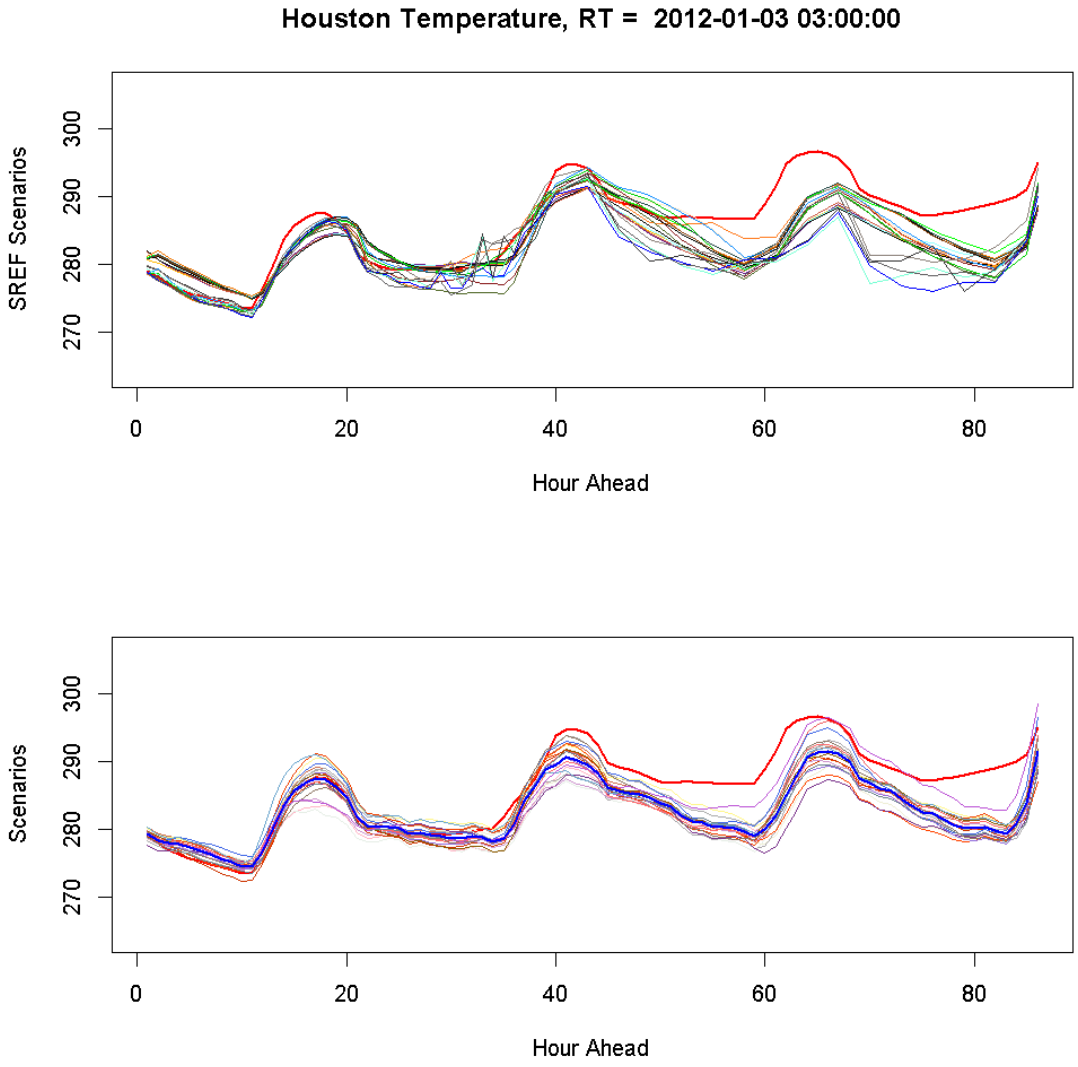

Figure 14: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.

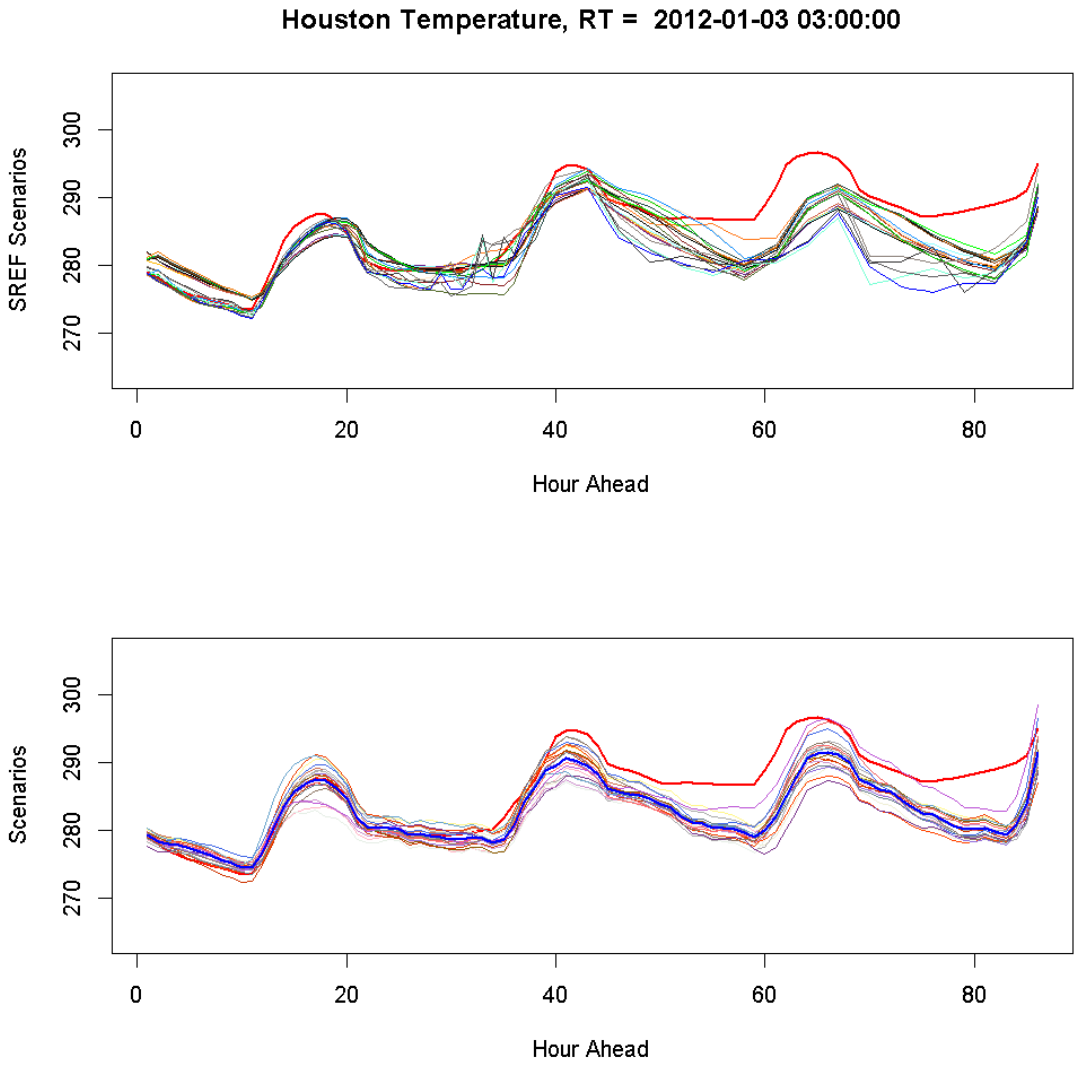

Figure 15: Top: SREF, Bottom: Statistical. The red line is the actual in both plots and the blue line in the bottom plot is the mean of the ensemble.